

Variable Structure Control in Active Queue Management for TCP with ECN*

Peng Yan

Dept. of Electrical Eng.
The Ohio State University
Columbus, OH 43210
yanp@ee.eng.ohio-state.edu

Yuan Gao

Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974-0636
yuangao@research.bell-labs.com

Hitay Özbay†

Dept. of Electrical & Electronics Eng.
Bilkent University
Bilkent, Ankara, Turkey TR-06533
ozbay@ee.eng.ohio-state.edu

Abstract

It has been shown that the TCP connections through the congested routers can be modeled as a feedback dynamic system. In this paper, we design a variable structure (VS) based control scheme in Active Queue Management (AQM) supporting explicit congestion notification (ECN). By analyzing the robustness and performance of the control scheme for the nonlinear TCP/AQM model, we show that the proposed design has good performance and robustness with respect to the uncertainties of the round-trip time (RTT) and the number of active TCP sessions, which are central to the notion of AQM. Implementation issues are discussed and ns simulations are provided to validate the design and compare its performance to other peer schemes' in different scenarios. The results show that the proposed design significantly outperforms the peer AQM schemes in terms of packet loss ratio, throughput and buffer fluctuation.

1 Introduction

Traffic congestion on the Internet is one of the major communication problems experienced by millions of users. Recently, for the purpose of alleviating congestion problem for IP networks and providing some notion of quality of service (QoS), AQM techniques have been proposed [5, 6, 7, 13, 14, 15]. Notably, the TCP connections through the congested routers can be modeled as a feedback dynamic system, where control theory based approaches can be used to analyze the network behavior and design new AQM schemes. We refer to [9, 12] for the details of the nonlinear TCP/AQM dynamic models. It is believed that the control system based analysis offers new insight into the AQM design. In [6], a control theoretic analysis was given for RED, which provided a more systematic and in-depth study on RED parameter tuning; and [7] developed a PI

controller as a new AQM scheme using linear system analysis. On the other hand, the AQM congestion control can be viewed as a convex optimization problem for the dynamic system [9, 10], where steady state properties and equilibriums are investigated. Taking the stability and performance indices into consideration, the above control theoretic AQM schemes achieve better performance in terms of high link utilization and low packet loss ratio.

Due to the challenging nature of nonlinearity in the TCP dynamics, most of the current results on the AQM analysis and design are based on linearized models which are valid only in the neighborhood of the equilibrium points ([6, 7, 10, 15]). To further complicate the situation, the TCP/AQM dynamics have time varying round-trip times (RTT) and uncertainties with respect to the number of active TCP sessions through the congested AQM router, which requires more robustness for the designed schemes.

In this paper, we introduce a robust variable structure (VS) based AQM scheme designed directly for the nonlinear TCP model, which to the best of our knowledge has not yet been employed to TCP/AQM system analysis and design. The motivations behind this work are (i) VS sliding mode control is robust and powerful for nonlinear systems, thus being well-suited for an AQM scheme; (ii) DropTail, a widely deployed AQM scheme, can be treated as a VS controller in the sense that its dropping policy is $p(t) = 1$ when $q(t) > q_{max}$, and $p(t) = 0$ when $q(t) \leq q_{max}$, where $p(t)$ is the dropping probability, $q(t)$ and q_{max} are the occupied buffer and the maximum buffer length respectively. Note that the undesirable behavior of DropTail (e.g. heavy queue fluctuation and high dropping rate), from the VS control perspective, is due to the fact that its binomial control doesn't fit TCP dynamics to enforce a stable sliding mode.

This paper is organized as follows. In Section 2, we discuss the TCP traffic dynamics in nonlinear differential equations and develop a VS based AQM controller supporting ECN. The ns simulation results are presented in Section 3 to validate our design, followed by the concluding remarks in Section 4.

*This work is supported by the National Science Foundation under grant number ANI-0073725.

†Hitay Özbay is on leave from The Ohio-State University

2 Variable Structure Control in AQM

2.1 Nonlinear TCP dynamics

In [9], a nonlinear dynamic model for TCP congestion control was derived, where the network topology was assumed to be a single bottleneck with M homogeneous TCP sources that share the bottleneck link and have roughly the same RTTs, but don't necessarily transverse the same path. For TCP with ECN, the AIMD behavior in congestion avoidance phase can be modeled as follows: each positive acknowledgment increases the value of congestion window $cwnd$ by $1/cwnd$ while each congestion indication (ECN) reduces the $cwnd$ by half, thus the expected change in congestion window is given by $\frac{1-p}{cwnd} - \frac{cwnd}{2}p$, where p is the marking probability. Aggregating the M TCP flows through one congested router, we have

$$\begin{cases} \dot{r}(t) &= \frac{M}{\tau^2(t)} - (\frac{M}{\tau^2(t)} + \frac{r^2(t)}{2M})p(t) \\ \dot{q}(t) &= r(t) - C_0, \end{cases} \quad (1)$$

and

$$\tau(t) = \frac{q(t)}{C_0} + T_p, \quad (2)$$

where $0 \leq p(t) \leq 1$ is the marking probability, $\tau(t)$ is the round trip time delay, $q(t)$ is the instantaneous queue length on the router, $r(t)$ is the incoming traffic rate per unit time, M is the number of the active TCP sessions, T_p is the propagation delay and C_0 is the link capacity. Note that (1) is a simplified model without considering the time delays in $r(t)$ and $p(t)$. The impact of this simplification will be discussed in Section 3.

To simplify the solution, instead of using (2), we assume $\tau(t)$ to be any continuous function independent of $q(t)$ with

$$T_p \leq T_0 \leq \tau(t) \leq T_1 \leq \frac{q_{max}}{C_0} + T_p, \quad (3)$$

where q_{max} is the buffer length of the router. Note that (3) captures its time varying nature and decouples $\tau(t)$ from $q(t)$. We further assume the number of active TCP connections to be uncertain, obeying

$$0 < M^- \leq M \leq M^+, \quad (4)$$

which is more reasonable in practice.

2.2 VS based AQM with ECN

The nonlinear TCP dynamics and the VS based AQM can be modeled as a feedback control system depicted in Figure 1, where the VS controller uses the queue and traffic incoming rate information to generate marking rate as congestion indication. Let $e_q = q(t) - q_d$ and denote $x_1 = e_q$,

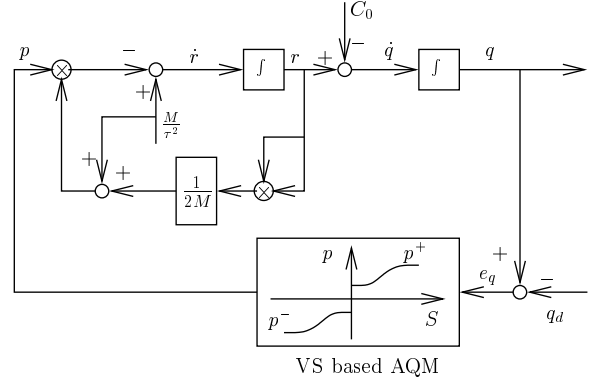


Figure 1. Aggregated dynamics of TCP and VS based AQM

$x_2 = \dot{e}_q$, where q_d is the desired queue length. We have

$$x_2 = \dot{e}_q = \dot{q}(t) = r(t) - C_0, \quad (5)$$

and the plant (1) can be described as

$$\begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{M}{\tau^2(t)} - (\frac{M}{\tau^2(t)} + \frac{(x_2 + C_0)^2}{2M})p(t) \end{cases} \quad (6)$$

Note that (6) is an affine nonlinear system and the equivalent control method (ECM) can be used to construct the VS control law. We select the sliding mode surface as

$$S(x, t) = k_S x_1 + x_2, \quad k_S > 0, \quad (7)$$

which corresponds to a linear combination of the queue length error and the error between incoming traffic rate and link capacity. Define

$$\begin{aligned} \bar{b}(x_2) &= \frac{\bar{M}}{\bar{\tau}^2} + \frac{(x_2 + C_0)^2}{2M} \\ b(x_2) &= \frac{M}{\tau(t)^2} + \frac{(x_2 + C_0)^2}{2M} \end{aligned} \quad (8)$$

where $\bar{M} := \frac{M^- + M^+}{2}$ and $\bar{\tau} := \frac{T_0 + T_1}{2}$ are the nominal values of M and $\tau(t)$ respectively. According to the sliding mode existence condition and the robustness criteria, we construct the VS controller as follows

$$\begin{aligned} p(x, t) &= \frac{\bar{M}}{\bar{\tau}^2 \bar{b}(x_2)} + (\alpha \frac{T_1}{\sqrt{2}} k_S |\frac{x_2}{x_2 + C_0}| + \delta) \text{sign}(S(x, t)) \\ &= \begin{cases} p^+(x, t) & \text{if } S(x, t) > 0 \\ p^-(x, t) & \text{if } S(x, t) < 0 \end{cases} \end{aligned} \quad (9)$$

where $\alpha \geq 1$, $\delta > 0$ are constants and

$$\begin{aligned} p^+(x, t) &= \frac{\bar{M}}{\bar{\tau}^2 \bar{b}(x_2)} + (\alpha \frac{T_1}{\sqrt{2}} k_S |\frac{x_2}{x_2 + C_0}| + \delta) \\ p^-(x, t) &= \frac{\bar{M}}{\bar{\tau}^2 \bar{b}(x_2)} - (\alpha \frac{T_1}{\sqrt{2}} k_S |\frac{x_2}{x_2 + C_0}| + \delta) \end{aligned} \quad (10)$$

It was shown in [17] that the VS controller (9) robustly stabilizes the nonlinear system (6) for all $\tau(t)$ and M obeying (3) and (4) respectively, if $\alpha \geq 1$ and

$$\delta > \gamma := \frac{1}{4} \left(\frac{\bar{M}}{M^-} + \frac{\bar{\tau}}{T_0} \right) \left(\frac{M^+ - M^-}{M^+ + M^-} + \frac{T_1 - T_0}{T_1 + T_0} \right). \quad (11)$$

The feedback system is robustly stable in the sense of Lyapunov, where the state trajectory continuously converges to the sliding manifold, and $q(t)$ exponentially converges to the desired queue length q_d with decay rate k_S once the trajectory reaches the sliding manifold.

Remark 1. For the purpose of fast convergence, we would like to pick k_S as large as possible. On the other hand, a larger k_S results in a larger magnitude of the marking probability $p(t)$ (from (9) and (10)), which increases the risk of saturation ($0 \leq p(t) \leq 1$). Practically, we choose k_S so that the two terms $k_S x_1$ and x_2 in (7) are balanced (e.g. $k_S = 10$ is feasible when $q_d = 100$ and $C_0 = 1000$).

3 Simulation Results

To validate the performance and the robustness of the proposed VS AQM, we implement it in *ns-2* and conduct a simulation study in different scenarios. Some representative AQM schemes, namely, DropTail, RED [5], REM [2] and PI [7], are also simulated for the purpose of comparison. Note that the sliding mode AQM (SMVS) in [15] is not considered for comparison because the simulations in [15] are questionable as discussed in [17].

3.1 Simulation Configuration

The dynamical behaviors of the above AQM schemes are simulated under a variety of network topologies and traffic sources. In particular, we consider the dumbbell network topology depicted in Fig. 2, where M TCP connections share a single bottleneck link. We assume that the TCP sources always have data to send. The links between the TCP sources and the router R_1 are 10Mbps links with a 40ms propagation delay, which are the same as those between the TCP sinks and the router R_2 . Router R_1 is connected to R_2 through a 10Mbps 20ms delay link. The maximum buffer size of each router is set to 300 packets (of size 1000 bytes). Meanwhile, we consider the network topology with multiple bottleneck links (Fig.3), where the maximum buffer of each router is 200 packets, the bandwidth and the propagation delay of each link are indicated in Fig.3 and each sender-receiver pair has n_{cross} TCP connections as cross traffic. In both scenarios, TCP-Reno is used as the transport agent.

The parameters used in VS based AQM (9) are: $\bar{M} = 100$, $\bar{\tau} = 0.2$, $K_s = 10$, $\alpha = 1$ and $\delta = 0.3$ (see

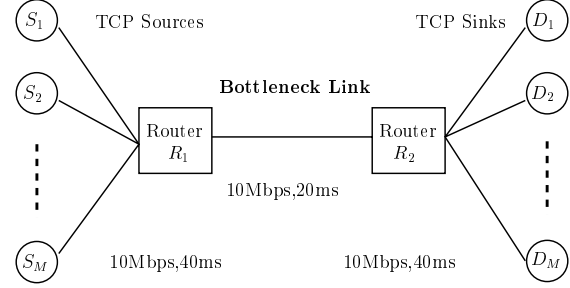


Figure 2. Dumbbell network topology

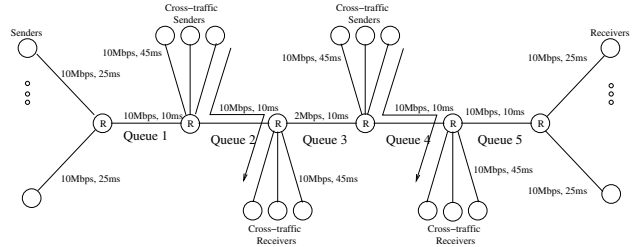


Figure 3. The network topology with multiple bottleneck links

example 1 in [17] for details). In PI AQM, we use the suggested parameter values $a = 1.822 \times 10^{-5}$ and $b = 1.816 \times 10^{-5}$ given in [7]. The desired queue length is set to $q_d = 150$ packets for VS control and PI control. The parameters of RED are set as recommended in <http://www.aciri.org/floyd/REDparameters.txt>. For REM, the parameters are set as $\alpha = 0.1$, $\gamma = 0.001$ and $\phi = 1.001$, which are recommended in [2]. In the following simulations, ECN is enabled for VS AQM, PI, REM and RED respectively, where $p(t)$ corresponds to the marking rate, and packet loss is observed only when the buffer overflows.

3.2 The Scenario of Single Bottleneck Topology

Performance Comparison of Different AQM Schemes In this experiment, we choose $M = 100$ in Fig. 2, which corresponds to 100 greedy FTP flows sharing the bottleneck link. The system response using the VS controller is depicted in Fig. 4, where the performance shows fast response and the stabilized queue size. Meanwhile, we repeat the same experiment using DropTail, RED, PI and REM respectively, and depict their instantaneous queues in Fig. 5.

As compared to the AQM schemes shown in Fig. 5, we clearly see that the VS AQM outperforms other schemes in

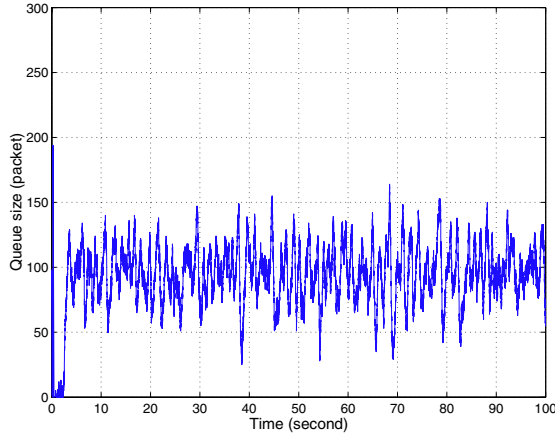


Figure 4. VS control

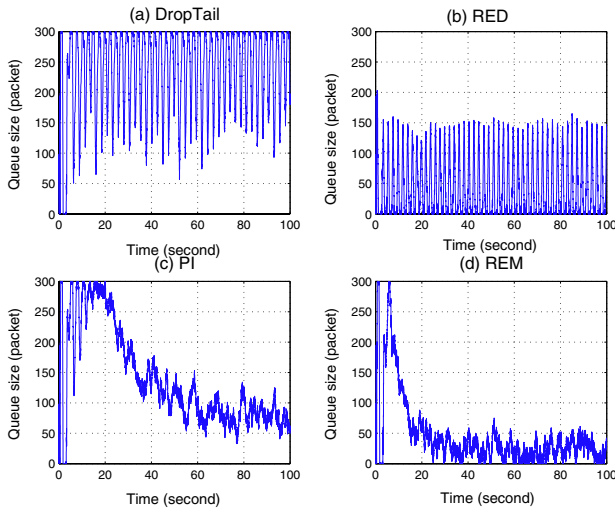


Figure 5. Other AQM schemes

terms of system stability and performance, which implies higher link utilization, lower packet loss ratio and smaller queue fluctuation. Although the PI controller is also a control theory based design which could regulate the queue to the desired value, its transient response is sluggish, which deteriorates its performance (Fig. 5 (c)).

Performance Under Dynamic Traffic Changes In this scenario, we provide some time-varying dynamics and investigate the performance of the VS controller and other representative schemes. We use 150 TCP connections at time $t = 0$. At time $t = 40$, 50 of the TCP connections stop transmitting data, and at time $t = 70$ they resume transmitting again. The queue evolution is depicted in Fig. 6. Note that PI (Fig. 6 (b)) and REM (Fig. 6 (c)) are not very robust with respect to such connection number varia-

tion, which result in heavy queue fluctuation during 40-70 second. Although RED (Fig. 6 (a)) is not very sensitive in this scenario, it tends to over-mark the incoming traffic so that the link utilization is degraded. As is evident from (Fig. 6 (d)), the VS controller is very robust against the variation of connections and keeps very good response even in the presence of such variations.

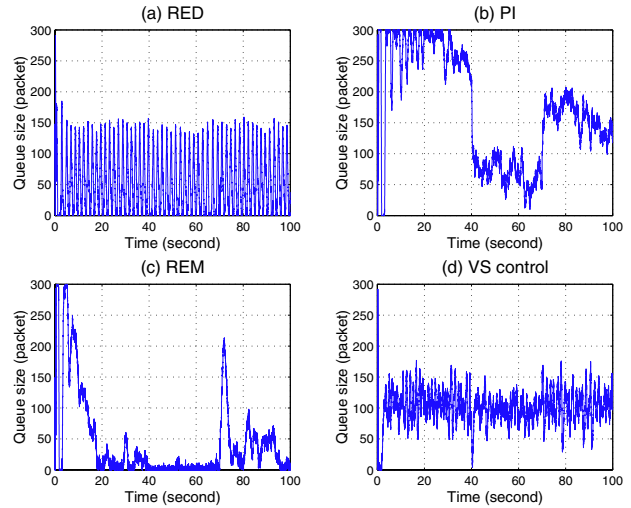


Figure 6. Queue evolution using RED, PI, REM and VS control

Robustness w.r.t. Number of TCP Connections The performance and robustness of the VS scheme are explored with respect to different TCP loads. We conduct simulations with the same setting as in Experiment 1, except that the number of connections varies from 50 to 250. Fig. 7 plots the average queue length (from 20 second to 100 second) for different AQM schemes. Correspondingly the link utilization and the packet loss ratio are depicted in Fig. 8 and Fig. 9 respectively.

It is observed that the VS controller can robustly stabilize the queue length around 100 packets. The average queue lengths of REM and RED vary slightly with respect to the flow number, while the average queue of PI blows up when the flow number increases, which is due to the inherent nature of PID control that system response is highly dependent on the system parameters.

From Fig. 8, we clearly see that the VS and PI controller have better link utilization than REM and RED, and the link utilization of REM and RED are sensitive to the variation of the flow numbers. Note that the packet loss ratio of PI increases with respect to higher loads (Fig. 9), which is not desirable for congestion control. REM has lower drop rate compared with PI, but its link utilization is not that good. Notice that in Fig. 9, the drop rate for RED is not plotted

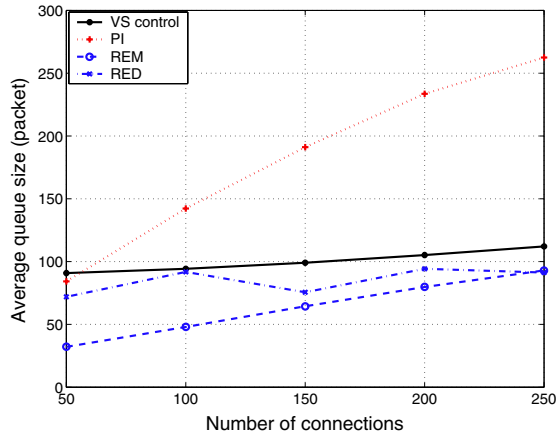


Figure 7. Average queue length

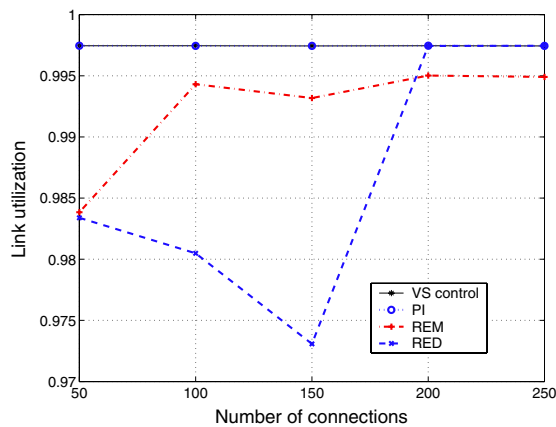


Figure 8. Link utilization

due to the fact that RED has a much higher drop rate in the range between 6% and 13% in this simulation. Compared with RED, REM and PI, the VS controller has much better performance in terms of robustly stabilized queue length, high link utilization and low packet loss ratio (in this experiment, the packet loss ratio for VS AQM is always 0).

The above analysis shows that the control theory based AQM designs, say, PI and VS AQM, have better link utilization by stabilizing the queue length to the desired values. Note that the system response (response time and overshoot) of PI is deteriorated by higher loads. Thus the high link utilization of PI is in the expense of high packet drop rate when the number of TCP flows increases. On the other hand, the VS controller keeps fast response, stability robustness with respect to a large range of variation of TCP flow numbers, which is well-suited as an AQM scheme.

Comparison with PI AQM under Higher TCP Loads It

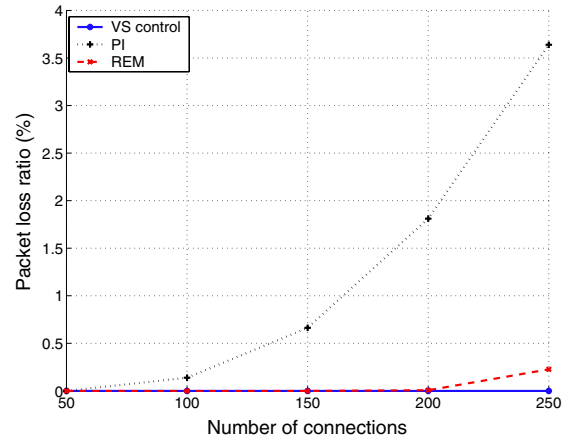


Figure 9. Packet loss ratio

has been shown in [7] that the PI controller has robustness with respect to the number of connections, although a larger number of connections results in a slower system response. In this experiment, we set the flow number to $M = 300$ and compare the system response of PI and the VS AQM. As shown in Fig. 10, the VS controller has very good transient response and very low overshoot. On the other hand, the PI controller exhibits much slower response and larger overshoot, which implies higher packet loss ratio and larger *RTT*.

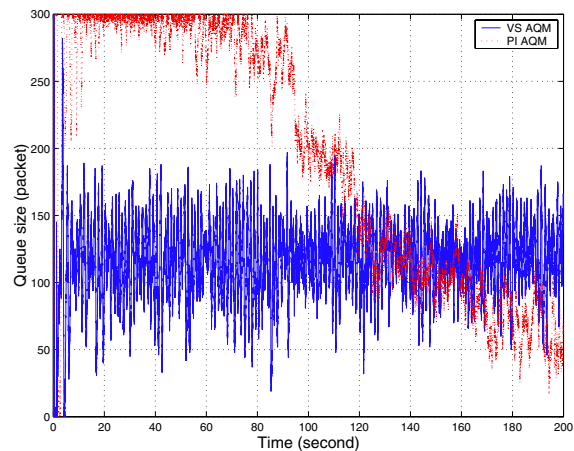


Figure 10. Comparison of PI and the VS controller

Performance in the Presence of Short-lived TCP Flows

In this experiment, we investigate the system performance in the presence of short-lived TCP flows. The single bottleneck topology we considered is depicted in Fig.2, where 120 greedy FTP flows and 50 short-lived TCP flows share

the link. Each short-lived TCP flow is configured to randomly turn ON/OFF in each 5s period. As shown in Fig.11, VS AQM has good robustness against the disturbance of such short-lived TCP flows, and can significantly improve the system performance.

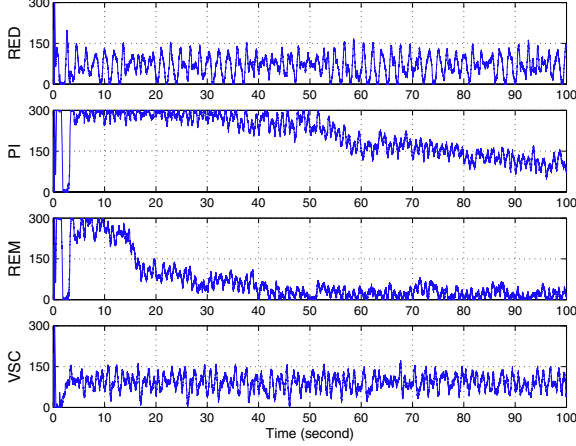


Figure 11. Performance in the presence of short-lived TCP flows

Robustness w.r.t. RTT As discussed in Section 2, the VS controller has very good robustness against the uncertainty of RTT , which is essential as an AQM scheme. In this experiment, we change the propagation delays in the network topology (Fig. 2) and evaluate the robustness of the VS controller. First, we set the propagation delay between router R_1 and R_2 to 10ms and the delays between the routers and the end hosts 2ms, which corresponds to a much smaller RTT than the nominal one. The regulated queues are depicted in Fig. 12 where the VS controller is still capable of robustly stabilizing the queue length. Note that for this scenario, RED (Fig. 12 (a)) also exhibits good performance.

Meanwhile, we also consider the scenario with a much larger RTT , where we set the propagation delay between R_1 and R_2 to 120ms and the delays between the routers and the end hosts 20ms. We repeat the simulation with the flow number $M = 100$ and obtain the system responses depicted in Fig. 13. As we can observe, the VS controller continues to exhibit good performance in the sense of queue stability and fast response, which outperforms other peer schemes.

3.3 The Scenario of Multiple Bottleneck Topology

Using the multiple bottleneck network topology depicted in Fig. 3, we study the behaviors of different AQM schemes in the presence of cross traffic. We set 300 TCP connections with sender at the left hand side and receivers at the right hand side, with 50 TCP flows ($n_{cross} = 50$) for each

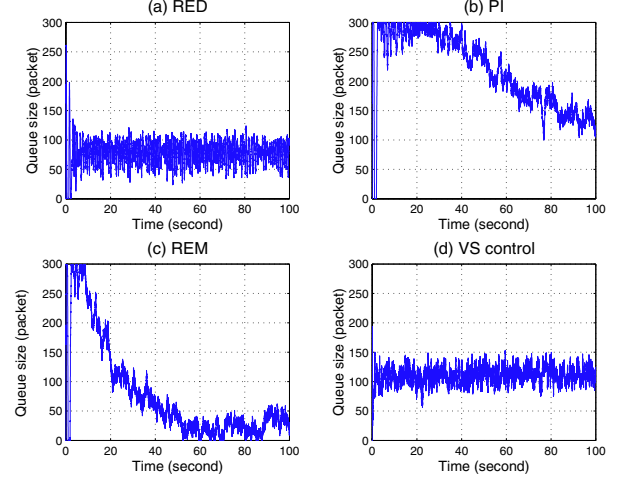


Figure 12. The scenario with a smaller propagation delay

cross traffic sender-receiver pair. The instantaneous queues of Queue 4 for different AQMs are depicted in Fig. 14. Note that Queue 2, 3 and 4 exhibit similar trends. Queue 1 and Queue 5 are almost empty, indicating that these two links are not bottleneck links. Similar results to Fig. 14 can be obtained under different TCP loads and different cross traffic loads.

Once again, the VS controller shows much better performance than other AQM methods. In fact, the performance of RED in this experiment is sensitive to the network configurations (e.g. TCP loads, cross traffic and propagation delays), which affirms coincidence to [4]. On the other hand, REM tends to mark too many packets and keeps too small a queue size, so that the link utilization is lower than other AQMs. Meanwhile, PI AQM suffers a sluggish transient behavior and makes the queue full in its transient period, which results in buffer overflow and packet losses.

Remark 3. Note that the VS controller has a steady state error ($\lim_{t \rightarrow \infty} q(t) - q_d$) of about 50 packets in the above scenarios. In fact, the VS control is designed for the delay-free nonlinear system (6) and the corresponding sliding manifold (7) is also delay-free. A more accurate model of the real network traffic is the delayed version of (6)

$$\begin{aligned} \dot{r}(t) &= \frac{Mr(t-\bar{\tau})}{\tau^2(t)r(t)} - \left(\frac{Mr(t-\bar{\tau})}{\tau^2(t)r(t)} + \frac{r(t)r(t-\bar{\tau})}{2M} \right) p(t-\bar{\tau}) \\ \dot{q}(t) &= r(t) - C_0, \end{aligned} \quad (12)$$

which was proposed in [9]. The presence of time delays results in $S(x, t) \neq 0$ in the system steady state, which causes the steady state error of the queue size.

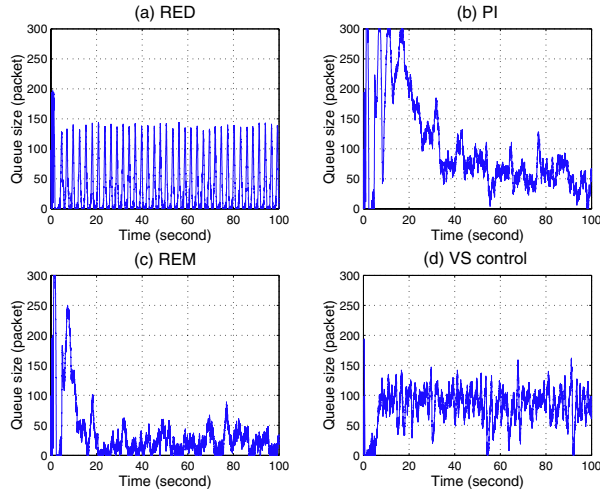


Figure 13. The scenario with a larger propagation delay

4 Concluding Remarks

In this paper, we developed a variable structure based AQM control scheme supporting ECN. We presented guidelines for designing the robust VS sliding mode controller directly for the nonlinear TCP dynamics. It was shown that the VS controller, from control theoretic point of view, has many desirable properties such as good robustness and fast system response. We also provided *ns* simulations in different scenarios to validate our results. The simulation experiments showed that the proposed AQM scheme performs better than a number of well-known AQM schemes in terms of packet loss ratio, link utilization and queue fluctuation. A challenging extension of this work is to consider VS control for the TCP model in the presence of time delays as shown in (12), which we are currently investigating.

References

- [1] UCB, LBNL, VINT network simulator. <http://www-mash.cs.berkeley.edu/ns/>.
- [2] S. Athuraliya, S. H. Low, V. H. Li, and Q. Yin. REM: Active queue management. *IEEE Network Magazine*, 15(3), May/June 2001.
- [3] B. Braden and et al. Recommendations on queue management and congestion avoidance in the Internet. In RFC-2309, April 1998.
- [4] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith. Tuning RED for web traffic. In *Proceedings of ACM SIGCOMM'00*, September 2000.
- [5] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4), August 1993.

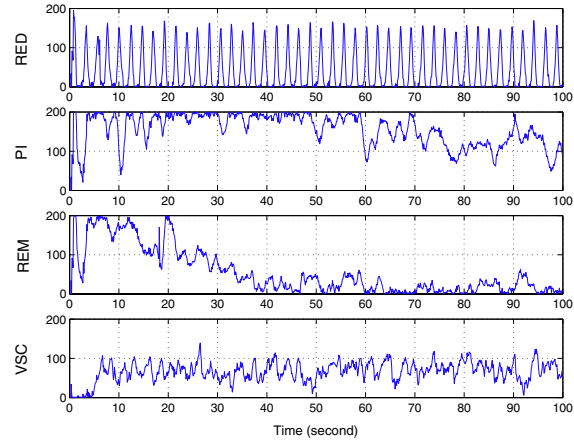


Figure 14. Evolution of Queue 4 (in packet) using RED, PI, REM and VS control

- [6] C. Hollot, V. Misra, D. Towsley, and W. Gong. A control theoretic analysis of RED. In *Proc. IEEE INFOCOM 2001*, April 2001.
- [7] C. Hollot, V. Misra, D. Towsley, and W. Gong. On designing improved controllers for AQM routers supporting TCP flows. In *Proc. IEEE INFOCOM 2001*, April 2001.
- [8] V. Jacobson and M. Karels. Congestion avoidance and control. In *Proceedings of ACM SIGCOMM'88*, August 1988.
- [9] F. Kelly. Mathematical modelling of the Internet. *Springer-Verlag, Berlin*, 2001.
- [10] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management. In *Proc. of ACM SIGCOMM 2001*, August 2001.
- [11] D. Lin and R. Morris. Dynamics of random early detection. In *Proceedings of ACM SIGCOMM'97*, September 1997.
- [12] V. Misra, W. Gong, and D. Towsley. A fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *Proceedings of ACM SIGCOMM'00*, September 2000.
- [13] T. Ott, T. Lakshman, and L. Wong. SRED: Stabilized RED. In *Proceedings of IEEE INFOCOM'99*, March 1999.
- [14] K. K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. In RFC-3168. *IETF*, September 2001.
- [15] F. Ren, X. Ying, Y. Ren, and X. Shan. A robust active queue management algorithm based on sliding mode variable structure control. In *Proceedings of IEEE INFOCOM'02*, June 2002.
- [16] V. Utkin. *Sliding Modes in Control and Optimization*. Springer Verlag, 1992.
- [17] P. Yan, Y. Gao, and H. Özbay. A variable structure control approach to active queue management for TCP with ECN. Submitted to *IEEE Transactions on Control Systems Technology*, 2003.